

## توضیح کد

در کلاس Textjustification متدی به نام justify داریم که اصلی ترین متد است. ابتدا یک سری متن را از کاربر گرفته و در ارایه words می ریزیم و هم چنین متغیر را width می گیریم. این متغیر تعیین می کند که طول رشته ای که کاربر وارد می کند بیشتر از width نباشد مثلاً اگر width=4 باشد برای کلمه abcde مقدار abcd را نگه میدارد در ادامه این فرایند توضیح داده می شود

ارایه cost برای حساب کردن هزینه قرار دادن کلمه است به این صورت عمل می کنیم که width را منهای طول فعلی هر کلمه می کنیم تا هزینش بدست بیاد در for بعد چک می کنیم اگر کلمه بعدی از کلمه فعلی بزرگتر بود میایم کلمه قبلی رو منهای کلمه فعلی می کنیم اگر کلمه بعدی طولانی تر بود میایم جایگزین قبلی می کنیم. حالا ارایه را چک می کنیم اگر خونه ای از ارایه هزینش منفی شد که یک مقدار منفی قرار می دهیم اگر مثبت بود هزینه را به توان دو رسونده و آن را به عنوان هزینه جدید در نظر می گیریم سپس کل ارایه را بررسی کرده تا کمترین هزینه را حساب کنیم. برای پیدا کردن کمترین هزینه را به این صورت عمل می کنیم که کپی از ارایه cost را در mincost می ریزیم

ابتدا متن را در buffer قرار می دهیم سپس چک می کنیم که از بزرگترین طول کلمه که همان مقدار width است آیا بیشتر شده است یا نه اگر کمتر بود اسپیس می زنیم اگر بیشتر بود میریم خط بعد.

تعداد اسپیس که بدست میاد در کل رشته پخش یعنی اگر 3 تا اسپیس لازم باشه در خط اول مثلاً یک اسپیس در خط بعدی دو اسپیس بستگی به شرایط دارد

## پیچیدگی

در دو for اول و دوم که تو در تو است و if هم که پیچیدگی ثابتی دارد پیچیدگی:  $2T(n^2)*t(1)=O(n^2)$

```
Enter number of elements:4      Enter number of elements:5
aaaaaaa                        aaaaaaaaaa
bb                               bbb
cc                               ccc
dddd                            dddd
min cost is 20                  min cost is 45

aaaaaaa bb                      aaaaaaaaaa
cc ddddd                       bbb ccc
                                dddd eee

Process finished with exit code 0 Process finished with exit code 0
```